

AN

May 2, 2001 3:14pm Page 1

t s6/7/all

6/7/1

IALOG(R) File 347:JAPIO
c) 2001 JPO & JAPIO. All rts. reserv.

6296384 **Image available**

GI PROGRAM PREPARING METHOD

UB. NO.: 11-237976 [JP 11237976 A]

PUBLISHED: August 31, 1999 (19990831)

INVENTOR(s): NUMANOI ATSUSHI
TAMURA KIYOAKI
SAKANISHI SENICHIRO
YOSHIKAWA SATOSHI
SAWARA HIROSHI

APPLICANT(s): HITACHI LTD

APPL. NO.: 10-056118 [JP 9856118]

FILED: February 20, 1998 (19980220)

ABSTRACT

PROBLEM TO BE SOLVED: To provide a common gateway interface(CGI) program preparing system with which the source of a CGI program for receiving/analyzing and processing an input character string and a hyper text* mark-up language(HTRML) for output can be *separately* generated/maintained and the execution performance of the generated CGI program can be made equal with that of an ordinary CGI program.

SOLUTION: In this program preparing method, a translator 404 is prepared for translating a prepared *HTML* 402 to a program source for outputting that *HTML* and while using that translator, the *HTML* to be an output of the CGI program is translated to a sub-program which can be called from the CGI program for performing processing excepting for the output of *HTML* such as receiving/analyzing and processing the input character string. Then, the CGI program prepared *separately* from the sub-program for performing the processing excepting for the output of *HTML* such as receiving/analyzing and processing the input character string and the sub-program translating the *HTML* are generated as logically one program.

COPYRIGHT: (C)1999,JPO

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平11-237976

(43)公開日 平成11年(1999)8月31日

(51)Int.Cl.
G 06 F 9/06

識別記号
530

F I
G 06 F 9/06

530 A
530 W

審査請求 未請求 請求項の数3 FD (全11頁)

(21)出願番号 特願平10-56118

(22)出願日 平成10年(1998)2月20日

(71)出願人 000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目6番地
(72)発明者 沼野井 淳
神奈川県横浜市戸塚区戸塚町5030番地 株式会社日立製作所ソフトウェア開発本部内
(72)発明者 田村 清朗
神奈川県横浜市戸塚区戸塚町5030番地 株式会社日立製作所ソフトウェア開発本部内
(72)発明者 坂西 宣一郎
神奈川県横浜市戸塚区戸塚町5030番地 株式会社日立製作所ソフトウェア開発本部内
(74)代理人 弁理士 矢島 保夫

最終頁に続く

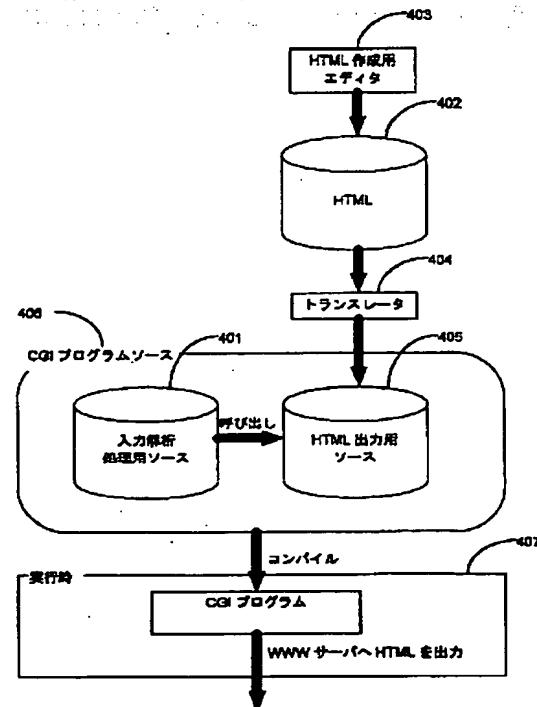
(54)【発明の名称】 CGI プログラム作成方法

(57)【要約】

【課題】入力文字列の受付・解析および処理を行うCGI プログラムのソースと出力用のHTMLとを別個に生成・保守でき、かつ、生成されるCGI プログラムの実行性能が通常のCGI プログラムと同等になるようなCGI プログラム作成方式を提案することを目的とする。

【解決手段】あらかじめ作成したHTMLを、それを出力するプログラムソースに変換するトランスレータを用意し、そのトランスレータを用いて、CGI プログラムの出力となるHTMLを、入力文字列の受付・解析および処理などHTMLの出力以外の処理を行うCGI プログラムから呼び出すことの出来るサブプログラムに変換する。その後、別々に作成した入力文字列の受付・解析および処理などHTMLの出力以外の処理を行うCGI プログラムと上記HTMLを変換したサブプログラムとを論理的に一つのプログラムとして生成する。

【効果】入力文字列の受付・解析および処理を行うCGI プログラムのソースとHTMLの作成効率・保守性の向上、およびそれらから生成したCGI プログラムの性能の向上を達成することができる。



【特許請求の範囲】

【請求項1】WWWサーバ上で動作するCGIプログラムを作成するCGIプログラム作成方法において、表示画面を決定するための静的な情報をHTMLとしてあらかじめ作成するステップと、該HTMLを、トランスレータにより、該HTMLと同イメージの表示画面を出力するプログラムソースに変換するステップと、該変換したプログラムソースと、出力の動的な制御などの処理を行うとともに上記変換したプログラムソースのプログラムを呼び出す別作成のプログラムソースとを一体化させて、一つのCGIプログラムを生成するステップとを備えたことを特徴とするCGIプログラム作成方法。

【請求項2】WWWサーバ上で動作するCGIプログラムを作成するCGIプログラム作成方法において、表示画面を決定するための情報を、変数および制御情報を表す拡張タグを挿入して、HTMLとしてあらかじめ作成するステップと、

該HTMLを、トランスレータにより、該HTMLに基づくイメージの表示画面を出力するプログラムソースに変換するステップであって、該変換の際には拡張タグの部分を該拡張タグに応じたプログラムシーケンスに変換するステップと、該変換したプログラムソースと、該変換したプログラムソースのプログラムを呼び出す別作成のプログラムソースとを一体化させて、一つのCGIプログラムを生成するステップとを備えたことを特徴とするCGIプログラム作成方法。

【請求項3】請求項1または2の何れか1つに記載のプログラム作成方法に係るプログラムを格納したことを特徴とする記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、WWWサーバ上で動作するCGIプログラム作成方法に関する。

【0002】

【従来の技術】近年、WWW (World Wide Web) 上での情報交換の需要は高まるばかりである。WWW上での基本的な情報のやりとりは、WWWサーバからWWWブラウザと呼ばれるクライアントに“ホームページ”と呼ばれるドキュメントを転送することにより行われる。ホームページとは、HTML (HyperText Markup Language) と呼ばれる言語で記述されたテキストファイルである。以下、HTMLにしたがって記述されたテキストデータを「HTML」と呼ぶものとする。WWWブラウザは、受け取ったHTMLを解析し、グラフィカルなページをディスプレイ上に出力する。

【0003】また、ユーザは、WWWブラウザに対して

10

20

30

40

50

文字列などの入力を行うこともできる。この場合、WWWサーバ側は、WWWブラウザから入力された文字列などを受け取り、その入力データに応じてデータベース検索などの処理をし、その処理の結果を示すHTMLをWWWブラウザに返す、というような処理が要求される。このような要求の実現方法として、CGI (Common Gateway Interface) を利用することが多い。

【0004】CGIは、WWWサーバのあるサーバマシン上のリソースとのインターフェースであり、WWWサーバとサーバマシン上のアプリケーションとの間での情報のやりとりの方法を規定している。CGIを利用してWWWサーバと情報をやりとりするアプリケーションプログラムを、CGIプログラムという。CGIプログラムは汎用的なプログラミング言語で作成することができる。上に述べたCGIプログラムについては、例えば、参考文献: William E. Weinman著「木下望監訳1997/1 インプレス発行「CGIブック」に詳しく記述されている。

【0005】図1に、公知例としてCGIプログラムを用いたクライアントとサーバとのやり取りを表す構成図を示す。WWWブラウザ101から、CGIプログラム実行要求とユーザによりブラウザ101に入力された情報102 (規定されたフォーマットの文字列にまとめられている) とを受け取ったWWWサーバ103は、CGIプログラム104を起動する。CGIプログラム104は、WWWサーバ103から渡された情報105 (ユーザ入力情報102と同じ) を元に処理を行い、HTML106を出力する。WWWサーバ103は、出力のHTML106を受け取り、WWWブラウザにHTML107 (HTML106と同じ) を送信する。

【0006】CGIプログラムは通常、以下の(1)～(3)のような処理を行う。

(1) WWWサーバから受け取った文字列の解析

WWWブラウザは、ユーザが入力した情報などを、制御コードを付加するなどして1つの文字列にまとめて、WWWサーバに送信する。WWWサーバはその文字列をそのままCGIプログラムに渡すため、CGIプログラムでは、制御コードを除去するなどしてその文字列を解析し、ユーザが入力した情報を変換する。

(2) ユーザの入力に応じた処理

文字列を解析した結果ユーザの入力情報内容が明らかになるので、それに応じた処理を行う。例えば、ユーザが“Tom”という個人名を入力した場合、Tomに関する情報をデータベースから検索するなどの処理を行う。

(3) HTML出力

上記(2)の処理に応じたHTMLを出力する。例えば、上記の例でTomの個人情報を検索して得た場合に、その個人情報を見やすいうように配置したHTMLとして出力するなどである。

【0007】一般に、HTMLの出力部分は、CGIプログラムを記述するプログラミング言語の標準出力構文内に埋められた構造になる。図2に、COBOLによるCGIプログラムのHTML出力部分のコード例201を示す。このコード例201において、“IDENTIFICATION DIVISION”の行およびその次の“PROGRAM-ID”的行は、見出し部である。次の“ENVIRONMENT DIVISION”的行は環境部であり、その次の“DATA DIVISION”行からの何行かはデータ部であり、“PROCEDURE DIVISION”行からこのプログラムの“END”的前までが構成部（手続き部）である。構成部の中のDISPLAY文が標準出力を行う文で、HTMLを出力するのに用いている。これら各部はCOBOLの仕様で記述されている。

【0008】このコード例201の手続き部の中段では、IF文で変数Iが“4”より大きいか否かを判別し、この条件に応じて出力を変化させている。具体的には、変数Iの大きさに応じて、出力する文字列の内容とその文字の大きさを変化させている。このように、CGIプログラムでは、条件によって出力するHTMLを変化させることもできる。

【0009】以上説明したCGIプログラムには以下の(1)～(3)のような問題点がある。

(1) CGIプログラムコードとして、図2のコード例201のように、HTMLを標準出力する構文を何行にもわたって記述しなければならない。これはCGIプログラム作成者にとって非常にわずらわしい作業である。

(2) 出力するHTMLを変更したい場合、HTMLを出力する構文がCGIプログラムコード内に分散することも多いため、HTMLのWWWブラウザによる出力のイメージを把握することが難しい。

(3) HTMLを出力する部分を修正したい場合、あるいはHTMLを出力する以外の部分を修正したい場合のどちらにおいても、同じCGIプログラムコードの修正が必要になることから、CGIプログラムコードにバグが入り込みやすくなり、保守が難しくなる。

【0010】このため、CGIプログラム作成において、入力文字列の受付・解析およびその結果を元に出力の制御などの処理を行う部分と、CGIプログラムが出力するHTMLの部分とを、別々に作成・保守できる環境が望ましい。

【0011】このような環境を実現する公知技術として、図3の方法が、参考文献：富士通PowerCOBOL97 COBOL「CGIサブルーチン 1.0 使用手引書」（電子マニュアル）で提案されている。この方法では、入力文字列の受付・解析および処理などのHTMLの出力以外の処理を行うCGIプログラムのソース301と、出力用のHTML302とを、別々に作成する。HTML作成には、作成した画面のイメージ

をHTMLに変換する公知のHTML作成用エディタ303を使用し、HTML作成労力を軽減することができる。CGIプログラムのソース301をコンパイルして生成されたCGIプログラム304は、実行時にHTML302を読み込んで出力する。この方法だと、入力文字列の受付・解析およびその結果を元に出力の制御などの処理を行うCGIプログラムのソースと出力用のHTMLとを別々に生成・保守することができ、上記の問題を解決することが出来る。

【0012】

【発明が解決しようとする課題】しかし、上記図3の方法には以下のようないくつかの問題点がある。すなわち、CGIプログラムが動作する度に

- (1) CGIプログラムがHTMLを読み込み
- (2) 読み込んだHTMLを出力し、WWWサーバ経由でWWWブラウザに表示する

という処理を行うため、通常のCGIプログラムと比較して、(1)のHTML読み込みの分実行速度が遅くなるという問題点が挙げられる。一般的にCGIプログラムはWWWブラウザからの要求毎に実行・終了するものであり、実行要求が頻繁に発生するようなアプリケーションの場合には、HTMLの読み込み処理が要求毎に行われることになるため、クライアントへのレスポンスが遅くなるという問題点がある。

【0013】そのため、入力文字列の受付・解析およびその結果を元に出力の制御などの処理を行うCGIプログラムのソースと出力用のHTMLとを別個に生成・保守でき、かつ、生成されるCGIプログラムの実行性能が通常のCGIプログラムと同等になるようなCGIプログラム作成環境が望まれる。

【0014】本発明の目的は、入力文字列の受付・解析およびその結果を元に出力の制御などの処理を行うCGIプログラムのソースと出力用のHTMLとを別個に生成・保守でき、かつ、生成されるCGIプログラムの実行性能が通常のCGIプログラムと同等になるようなCGIプログラム作成方法を実現することにある。

【0015】

【課題を解決するための手段】上記目的を達成するため、請求項1に係る発明は、WWWサーバ上で動作するCGIプログラムを作成するCGIプログラム作成方法において、表示画面を決定するための静的な情報をHTMLとしてあらかじめ作成するステップと、該HTMLを、トランスレータにより、該HTMLと同イメージの表示画面を出力するプログラムソースに変換するステップと、該変換したプログラムソースと、出力の動的な制御などの処理を行うとともに上記変換したプログラムソースのプログラムを呼び出す別作成のプログラムソースとを一体化させて、一つのCGIプログラムを生成するステップとを備えたことを特徴とする。

【0016】請求項2に係る発明は、WWWサーバ上で

動作するCGIプログラムを作成するCGIプログラム作成方法において、表示画面を決定するための情報を、変数および制御情報を表す拡張タグを挿入して、HTMLとしてあらかじめ作成するステップと、該HTMLを、トランスレータにより、該HTMLに基づくイメージの表示画面を出力するプログラムソースに変換するステップであって、該変換の際には拡張タグの部分を該拡張タグに応じたプログラムシーケンスに変換するステップと、該変換したプログラムソースと、該変換したプログラムソースのプログラムを呼び出す別作成のプログラムソースとを一体化させて、一つのCGIプログラムを生成するステップとを備えたことを特徴とする。

【0017】請求項3に係る発明は、請求項1または2のプログラム作成方法に係るプログラムを格納した記録媒体に係るものである。

【0018】

【発明の実施の形態】以下、図面を用いて、本発明の実施の形態を説明する。ここでは、本発明の好適用例として、プログラミング言語COBOLを使用したプログラム開発環境を取り上げて説明する。

【0019】図4に、本発明の実施の形態の構成図を示す。図4において、入力文字列の受付・解析およびその結果を元に出力の制御などを行うCGIプログラムのソース401と出力用のHTML402とをあらかじめ作成するところまでは上述の図3の公知の手法と同じである。HTML402は、画面イメージからHTMLに変換するような公知のHTML作成用エディタ403を使用して作成することができる。トランスレータ404は、HTML402をHTML出力用CGIプログラムソース405に変換する。このトランスレータ404により、プログラムソース405は、入力文字列の受付・解析およびその結果を元に出力の制御などを行うCGIプログラム401から呼び出されるサブプログラムとして生成される。ユーザが入力文字列の受付・解析および処理を行うプログラム401に、トランスレータ404によって変換されたサブプログラム405を呼び出す構文を加えることにより、これらのプログラムは論理的に一つのプログラム406となる。このように、HTMLが自動的にプログラミング言語に変換されてCGIプログラム内に埋め込まれて、図2の通常のCGIプログラム201と同様に動作するソース406が生成される。そのため、このCGIプログラムソース406をコンパイルした実行形式のCGIプログラムは、407に示すように、プログラム実行中にファイル読み込みを行う必要はなく、通常のCGIプログラムと同等の実行性能を出すことができる。

【0020】図5に、トランスレータ404の動作を示す。このトランスレータは、
1. HTMLのファイル名を元にしたプログラム名の作成

2. HTMLを読み込み、出力構文に変換
という処理を行う。

【0021】まず、ステップ501で見出し部（IDENTIFICATION DIVISION）をファイル出力する。COBOLでは見出し部にプログラム名（PROGRAM-ID）を記述しなければならない。ここでは、プログラム名としてHTMLのファイル名を使用するものとする。入力文字列の受付・解析およびその結果を元に出力の制御などを行うCGIプログラム

10 は、このプログラム名によって、このCGIプログラム（トランスレータによりHTMLを変換したプログラム）を呼び出すことになる。次に、環境部・データ部・構成部などの必要な構文をファイル出力する（ステップ502）。その後、HTMLを行単位で読み取り、それを出力するDISPLAY文（COBOLの標準出力構文）に変換し、ファイルに出力する。この処理を、HTMLファイルの最後尾に辿りつくまで繰り返し行う（ステップ503）。最後に、プログラム終了の構文をファイルに出力する（ステップ504）。以上により、HTML

20 ファイルから、そのHTMLを出力するプログラムソースが得られたことになる。

【0022】図6から図9を参照して、本実施の形態の動作例を説明する。この例は、図6の601および602の2つのHTMLを、ユーザからの入力に応じて表示しわける動作例である。

【0023】まず、図6のCGIプログラムの出力イメージ601と602を公知のHTML作成用エディタで作成する。CGIプログラムは、ユーザからの入力に応じて出力イメージ601と602のどちらかを表示する
30 という処理を、内部に持つFLGというフラグ変数の値を参照して行うものとする。例えば、ユーザに性別を問い合わせ、男性だった場合にはFLG=1、女性であったらFLG=2というように設定される。出力イメージ601はCGIプログラム内でFLG=1の場合に出力され、出力イメージ602はFLGが1以外の場合に出力される。

【0024】図7は、図6の出力イメージ601および602を表示するHTMLである。図6の画面イメージを公知のHTML作成用エディタで作成し、図7のHTMLに変換することも出来る。WWWブラウザがHTML701を解釈した結果、出力イメージ601が表示される。また、WWWブラウザがHTML702を解釈した結果、出力イメージ602が表示される。

【0025】図8に、HTMLがトランスレータによって変換されたCGIプログラムの結果の例を示す。801はHTML701がトランスレータによって変換されたCOBOLソースの結果の例であり、802はHTML702がトランスレータによって変換されたCOBOLソースの結果の例である。図5のステップ501の処理により、HTML701のファイル名（MALE.h

tm1) からプログラム名が生成されている(803)。また、ステップ503の処理により、HTML701の内容が、COBOLの標準出力構文であるDISPLAY文に変換されている(804)。トランスレータにより、HTML702がCOBOLソース802に変換される際にも同様の処理が行われる。

【0026】図9に、図8のプログラムを呼び出すCGIプログラムソースの例を示す。COBOLソース901では、フラグFLGが1の場合には、HTML701を出力するサブプログラムMALE801を呼び出す(902)。サブプログラム名MALEは、HTMLファイル701の名称から決定されたプログラム名である。また、COBOLソース901では、フラグFLGが1以外の場合には、HTML702を出力するサブプログラムFEMALE802を呼び出す(903)。801、802および901の3つのCGIソースは論理的に一つのCGIプログラムであり、一つの実行形式ファイルにコンパイルすることができる。

【0027】次に、本発明の第2の実施の形態を説明する。前出の例では、出力を動的に変更するために2つのHTMLを用意している。一方、HTMLに変数や制御構造を示すHTMLの仕様にない拡張タグを挿入することによって、一つのHTMLで動的な出力の変更に対応するような手法が考案されている。このHTMLに拡張タグを挿入する手法については、例えば参考文献：大澤文孝著 1997/4 SOFTBANK発行「Webアプリケーション構築ガイド」に記述されている。本発明の第2の実施の形態として、このような、拡張タグを挿入したHTML(以下、拡張HTMLと呼ぶ)を使用した場合の動作例を示す。発明の構成図は図4と同じであるが、402に示すHTMLが拡張されている点と、404のトランスレータがこれに対応して拡張されている点が異なる。

【0028】HTMLに挿入する拡張タグとしては、変数を示すものや、

- ・ IF～ELSE～ENDIF (分岐構造)
- ・ REPEAT～ENDREPEAT (くり返し構造)

のように、基本的な制御構造を表現するものを用意するのが一般的である。

【0029】図10は、図6の出力イメージを得るために拡張HTMLの例である。\$%IF%\$、\$%ELSE%\$、\$%END-IF%\$という、条件分岐の制御構造を表すタグがHTMLに挿入されている(1001)。このHTMLは、他のプログラムによって処理されたときに、FLGの値が1である場合には図6の601のように“male”というタイトルで“I'm a man”という文字列を出力するHTMLとして機能し、FLGの値が1でない場合には図6の602と同様に“female”というタイトルで“I'm a woman”という文字列を出力するHTMLとして機能

する。FLGの値は、入力文字列の受付・解析およびその結果を元に出力の制御などを行うCGIプログラム側で決定される。

【0030】図11に、図10を拡張HTML用に拡張したトランスレータによって変換したCGIプログラムを示す。拡張したトランスレータでは、図10のHTMLに挿入されている拡張タグ内で使用されているフラグFLG1002を引数として受け取るようにプログラムを生成する(1101)。また、トランスレータは、図10の1001の制御構造を、図11の1102のように変換する。

【0031】図12に、図11のプログラムを呼び出すCGIプログラムソースの例を示す。FLGを引数として図11のプログラムを呼び出している(1201)。

【0032】図11と図12の2つのCGIソースは論理的に一つのCGIプログラムであり、一つの実行形式ファイルにコンパイルすることができる。

【0033】図13に、拡張HTML用のトランスレータの処理方式を示す。「HTMLを一行読み取る」ところまでは図5の第1の実施の形態と同じである。HTMLを読み取ったのち、その読み取った一行が拡張タグか否かを判別する。拡張タグでないときは、図5と同様にDISPLAY分を出力すればよい。拡張タグを検出した場合は、拡張タグの示す変数の処理や、制御構造をCGIプログラムに変換する。例えば、拡張タグ\$%IF%\$を検出したときには、IF文を出力し(ステップ1301)、拡張タグ\$%IF%\$を検出するまで、HTMLを一行読み取って(ステップ1303)、%\$ELSE\$%ならELSEを出力し、そうでなければ読み取ったHTMLを出力するDISPLAY文を出力する処理(ステップ1304)を繰り返す。他の拡張タグの場合も同様である。このように、変数や制御構造を含むようにタグを拡張したHTMLに対しても、そのタグに対応するCGIプログラムソースを生成する方法を定義することでCGIプログラムソースに変換することができる。

【0034】以上のようにして、本発明の目的を達成するための機能を実現するプログラムを作成することができる。また、プログラムを各種媒体に格納するための公

40 知技術を使用して、これらのプログラムを格納した媒体を作成することが出来る。

【0035】なお、ここでは特にCOBOL言語でCGIプログラムを作成する例を示したが、HTMLや拡張HTMLを他言語のソースに変換するトランスレータを用意することにより、C言語やC++言語などの他の汎用プログラミング言語に対しても本発明の適用は可能である。

【0036】

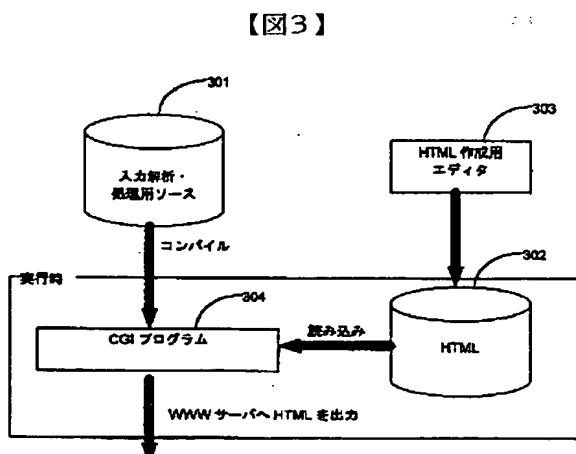
【発明の効果】以上説明したように、本発明によれば、CGIプログラム作成工程において、入力文字列の受付

・解析および処理を行うCGIプログラムのソースと出力用のHTMLとを分離するとともに、これらを1つのソースプログラムとして一体化することにより、それぞれの作成効率・保守性の向上を図ることができ、さらに生成されたCGIプログラムの性能の向上を実現することができる。

【図面の簡単な説明】

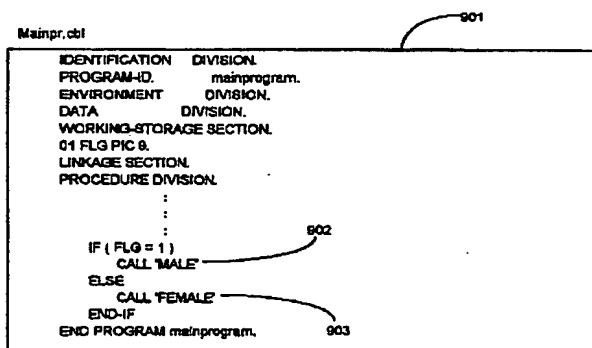
- 【図1】CGIプログラムの動作を示す構成図
- 【図2】CGIプログラムのHTML出力コード例を示す図
- 【図3】CGIプログラム作成の公知手法を示す図
- 【図4】第1の実施の形態の構成図
- 【図5】トランスレータの動作を示す図
- 【図6】作成するCGIプログラムで出力する画像イメージの例を示す図
- 【図7】画像イメージを表すHTMLの例を示す図
- 【図8】HTMLから変換したCGIプログラムの例を示す図
- 【図9】HTMLから変換したCGIプログラムを呼び出すプログラムの例を示す図
- 【図10】画像イメージを表す拡張HTMLの例を示す図
- 【図11】拡張HTMLから変換したCGIプログラムの例を示す図
- 【図12】拡張HTMLから変換したCGIプログラムを呼び出すプログラムの例を示す図
- 【図13】拡張HTMLをCGIプログラムに変換するトランスレータの動作例を示す図

【符号の説明】

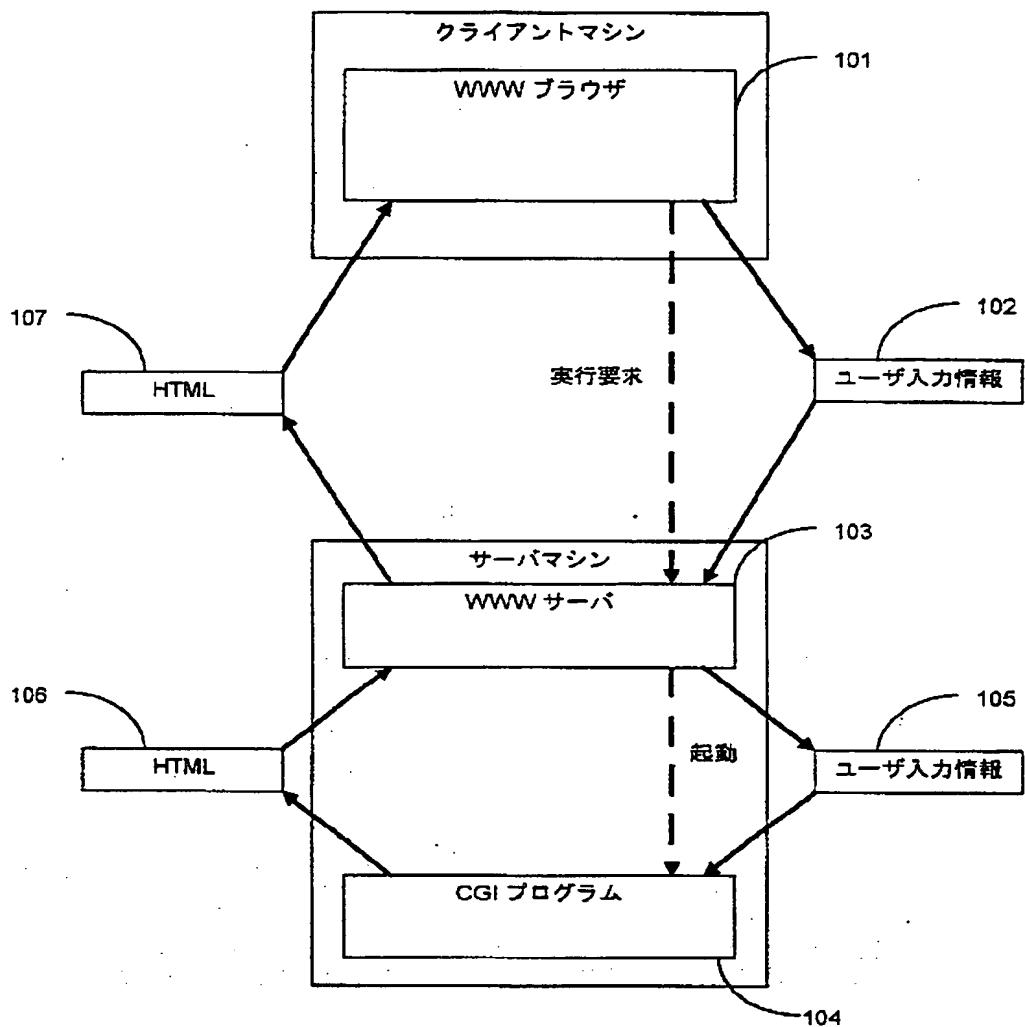


10 1 0 1…WWWブラウザ、1 0 2…ユーザ入力情報、1 0 3…WWWサーバ、1 0 4…CGIプログラム、1 0 5…ユーザ入力情報、1 0 6…HTMLファイル、1 0 7…HTMLファイル、2 0 1…HTML出力コード、3 0 1…入力解析・処理用ソースファイル、3 0 2…HTMLファイル、3 0 3…HTML作成用エディタ、3 0 4…CGIプログラム、4 0 1…入力解析・制御用ソースファイル、4 0 2…HTMLファイル、4 0 3…HTML作成用エディタ、4 0 4…トランスレータ、4 0 5…HTML出力用ソースファイル、4 0 6…CGIプログラムソース、4 0 7…CGIプログラム実行時の構成、5 0 1…見出し部ファイル出力部、5 0 2…環境部・データ部・構成部ファイル出力部、5 0 3…HTML変換・ファイル出力部、5 0 4…プログラム終わり見出しファイル出力部、6 0 1…WWWブラウザ出力イメージ、6 0 2…WWWブラウザ出力イメージ、7 0 1…HTMLコード、7 0 2…HTMLコード、8 0 1…HTMLを出力するCGIプログラムソース、8 0 2…HTMLを出力するCGIプログラムソース、8 0 3…プログラム名、8 0 4…CGIプログラムソース中のHTML出力部分、9 0 1…入力解析・制御用ソース、9 0 2…HTMLより変換されたCGIプログラム呼び出し部、9 0 3…HTMLより変換されたCGIプログラム呼び出し部、1 0 0 1…制御を表す拡張タグ例、1 0 0 2…変数を表す拡張タグ例、1 1 0 1…変数を表す拡張タグの変換例、1 1 0 2…制御を表す拡張タグの変換例、1 2 0 1…拡張HTMLより変換されたCGIプログラム呼び出し部、1 3 0 1…制御を表す拡張タグ変換部の動作例。

【図9】



【図1】



【図12】

```

IDENTIFICATION DIVISION.
PROGRAM-ID.      Mainprogram2.
ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.
  01 FLG PIC 9.
LINKAGE SECTION.
PROCEDURE DIVISION.
  :
  CALL 'IFSAM' USING FLG
END PROGRAM Mainprogram2.

```

1201

【図2】

```

201
IDENTIFICATION DIVISION.
PROGRAM-ID. mainprogram.
ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.
01 FLO PIC 9.
LINKAGE SECTION.
PROCEDURE DIVISION.

:
DISPLAY '<HTML>'
DISPLAY '<HEAD>'
DISPLAY '<TITLE>male</TITLE>'
DISPLAY '<HEAD>'
DISPLAY '<BODY>'
DISPLAY '<BR>'
DISPLAY '<H1>'

WITH NO ADVANCING
PERFORM VARYING I FROM 1 BY 1 UNTIL I>10
DISPLAY I WITH NO ADVANCING
DISPLAY '</H1>'

IF I>4
  DISPLAY '<H3>'
  DISPLAY "more than 4."
  DISPLAY '<H3>'

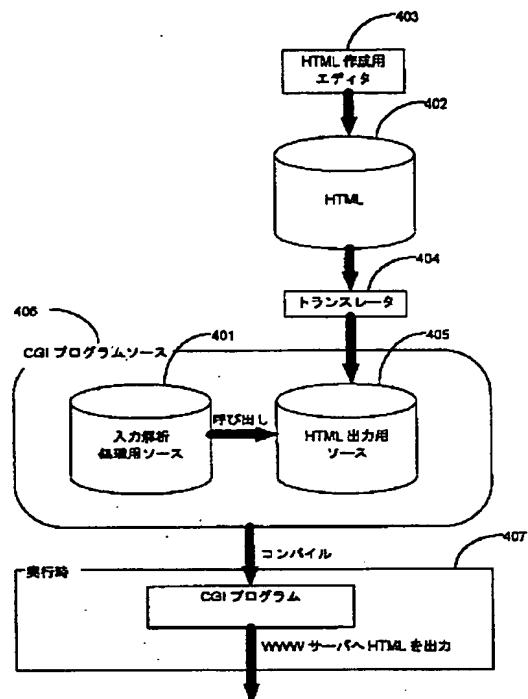
ELSE
  DISPLAY '<H2>'
  DISPLAY "less than 5.</H2>"
  DISPLAY '<H2>'

ENDIF
END-PERFORM
DISPLAY '<BR>'
DISPLAY '<BODY>'
DISPLAY '</HTML>'

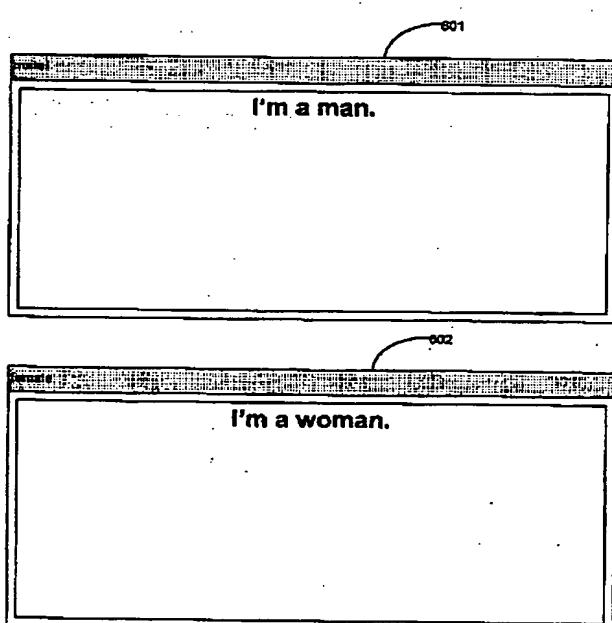
END PROGRAM mainprogram.

```

【図4】



【図6】



【図8】

```

801
IDENTIFICATION DIVISION.
PROGRAM-ID. MALE.
ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
PROCEDURE DIVISION USING BY REFERENCE CGI-ARGUMENT-LIST.
FORM-BEGIN.
DISPLAY '<HTML>'
DISPLAY '<HEAD>'
DISPLAY '<TITLE>male</TITLE>'
DISPLAY '<HEAD>'
DISPLAY '<BODY>'
DISPLAY '<BR>'
DISPLAY '<H1 ALIGN="center">I'
  WITH NO ADVANCING
  DISPLAY 'I'm a man.</H1>'
DISPLAY '<BR>'
DISPLAY '<BODY>'
DISPLAY '</HTML>'

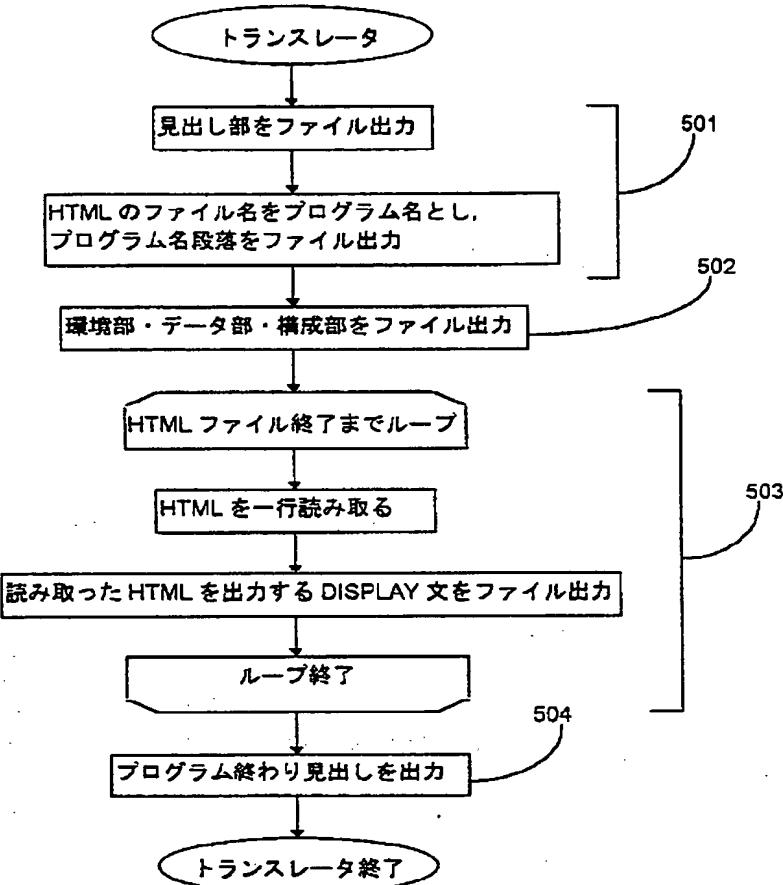
802
EXIT PROGRAM.
END PROGRAM MALE.

803
IDENTIFICATION DIVISION.
PROGRAM-ID. FEMALE.
ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
PROCEDURE DIVISION USING BY REFERENCE CGI-ARGUMENT-LIST.
FORM-BEGIN.
DISPLAY '<HTML>'
DISPLAY '<HEAD>'
DISPLAY '<TITLE>female</TITLE>'
DISPLAY '<HEAD>'
DISPLAY '<BODY>'
DISPLAY '<BR>'
DISPLAY '<H1 ALIGN="center">I'
  WITH NO ADVANCING
  DISPLAY 'I'm a woman.</H1>'
DISPLAY '<BR>'
DISPLAY '<BODY>'
DISPLAY '</HTML>'

804
EXIT PROGRAM.
END PROGRAM FEMALE.

```

【図5】



【図10】

IFSAM.HTML

```

<HTML>
<HEAD>
<TITLE>
$%IF ( FLG = 1 )%$ 1002
  male
$%ELSE%$ 1001
  female
$%END-IF%$ 
</TITLE>
<HEAD>
<BODY>
<BR>
<H1 ALIGN="center">
$%IF ( FLG = 1 )%$ 1002
  I'm a man.
$%ELSE%$ 1001
  I'm a women.
$%END-IF%$ 
</H1>
<BR>
<BODY>
</HTML>
  
```

【図7】

MALE.html

```
<HTML>
<HEAD>
<TITLE>male</TITLE>
</HEAD>
<BODY>
<BR>
<H1 ALIGN="center">I'm a man.</H1>
<BR>
</BODY>
</HTML>
```

701

FEMALE.html

```
<HTML>
<HEAD>
<TITLE>female</TITLE>
</HEAD>
<BODY>
<BR>
<H1 ALIGN="center">I'm a woman.</H1>
<BR>
</BODY>
</HTML>
```

702

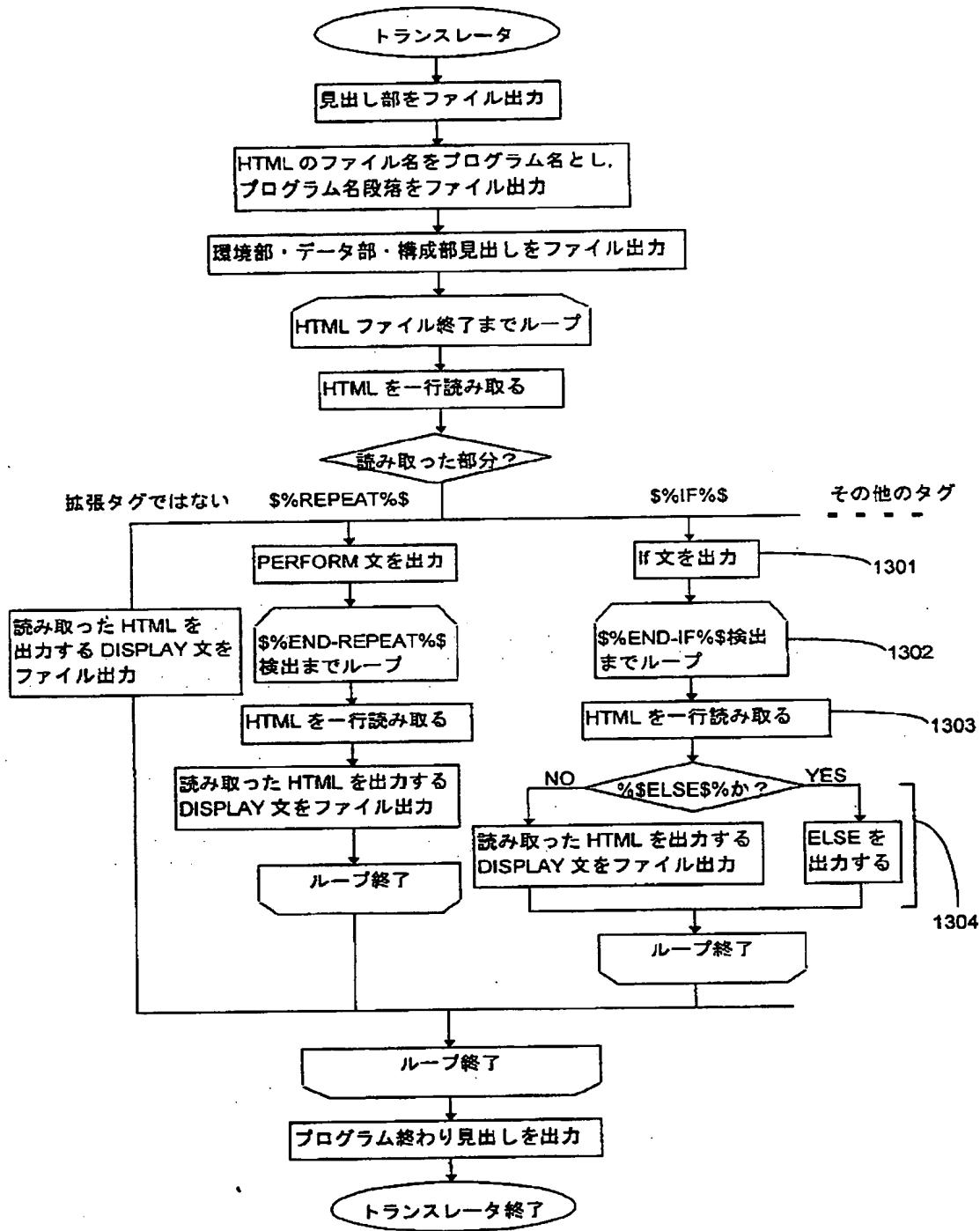
【図11】

```
IDENTIFICATION DIVISION.
PROGRAM-ID. IFSAM.
ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
01 FLG PIC 89(8) COMP.
PROCEDURE DIVISION.
USING FLG.
DISPLAY '<HTML>' 1101
DISPLAY '<HEAD>' 1102
DISPLAY '<TITLE>'
```

```
IF (FLG = 1)
  DISPLAY 'male'
ELSE
  DISPLAY 'female'
END-IF
DISPLAY '</TITLE>' 1101
DISPLAY '</HEAD>' 1102
DISPLAY '<BODY>' 1101
DISPLAY '<BR>' 1102
DISPLAY '<H1 ALIGN="center">' 1101
```

```
IF (FLG = 1)
  DISPLAY "I'm a man."
ELSE
  DISPLAY "I'm a woman."
END-IF
DISPLAY '<H1>' 1101
DISPLAY '<BR>' 1102
DISPLAY '<BODY>' 1101
DISPLAY '<HTML>' 1102
EXIT PROGRAM.
END PROGRAM IFSAM.
```

【図13】



フロントページの続き

(72)発明者 吉川 聰

神奈川県横浜市戸塚区戸塚町5030番地 株式会社日立製作所ソフトウェア開発本部内

(72)発明者 佐原 宏史

神奈川県横浜市戸塚区戸塚町5030番地 株式会社日立製作所ソフトウェア開発本部内